DLR-IB-AS-BS-2022-35

H- and P-Multigrid Methods for Higher-Order Discontinuous Galerkin Discretizations

Forschungsbericht

Malte Wegener

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Bericht des Instituts für Aerodynamik und Strömungstechnik
Report of the Institute of Aerodynamics and Flow Technology

DLR-IB-AS-BS-2022-35

H- and P-Multigrid Methods for Higher-Order Discontinuous Galerkin Discretizations

Malte Wegener

Herausgeber:
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Aerodynamik und Strömungstechnik
Lilienthalplatz 7, 38108 Braunschweig

ISSN 1614-7790

Stufe der Zugänglichkeit: 1
Braunschweig, im März 2022

Institutsdirektor:
Prof. Dr.-Ing. habil. C.-C. Rossow

Verfasser:
Malte Wegener

Abteilung: Center of Computer Applications in Aerospace Science and Engineering

Abteilungsleiter:
Prof. Dr. S. Görtz

Der Bericht enthält:
38 Seiten
12 Bilder
7 Tabellen
16 Literaturstellen

Abstract

A non-linear multigrid method for higher-order discontinuous Galerkin discretizations was developed. Discontinuous Galerkin methods allow for two different types of creating coarse levels. The first is an agglomeration based coarsening and the second is a reduction of the polynomial order of the ansatz functions. These h and p multigrid methods were furthermore combined in a nested multigrid method. The different multigrid choices were analysed on different test cases ranging from simple inviscid flow over a smooth bump to a multi-element high lift airfoil. It was found, that multigrid methods can reduce the computational time for high-order discontinuous Galerkin methods significantly. Furthermore, they exhibit increased robustness to the initial CFL number used for the computation. It was observed, that the treatment of viscous terms on agglomerated meshes using the BR2 scheme needs special attention and possible reasons for this are discussed.

In addition to non-linear multigrid methods, an assessment of a linear multigrid method as a preconditioner for linear systems arising from finite volume discretizations was performed. In this assessment, the preconditioner was found to be significantly stronger than a classical Jacobi preconditioner, which resulted in a reduction of runtime. Furthermore, the possibility for optimization of these methods was shown for the example of shifting the relative workload from the finest to coarser level, which proved to be beneficial in terms of run time. A comparison between a uniform residual restrictor and a volume-weighted restrictor was performed, however, no benefit was found using the volume-weighted restrictor.

Kurzfassung

Ein nichtlineares Mehrgitter-Verfahren für Discontinuous-Galerkin-Verfahren höherer Ordnung wurde entwickelt. In Discontinuous-Galerkin-Diskretisierungen kann ein grobes Level auf zwei unterschiedliche Weisen gebildet werden. Zum einen kann das Netz durch eine Agglomeration vergröbert werden, zum anderen kann der Polynomgrad der Ansatzfunktionen reduziert werden. Diese h- und p- Mehrgitter-methoden wurden in einem geschachtelten Mehrgitter kombiniert und an einer Reihe von numerischen Testfällen analysiert. Die Betrachtung erfolgte an reibungsfreien Testfällen uber einfache Geomtrien bis zu viskosen Strömungen über ein Drei-Element-Tragflügelprofil. Es wurde gezeigt, dass die betrachteten Mehrgitter-Verfahren die Laufzeit der Berechnungen verringern konnten. Weiterhin wurde eine Steigerung der Robustheit im Bezug auf die initiale CFL-Zahl beobachtet. Es zeigt sich, dass die Behandlung der viskosen Flüsse mit dem BR2-Schema besondere Aufmerksamkeit benötigt.

Weiterhin wurde ein lineares Mehrgitter als Vorkonditionierer im Kontext einer Finite-Volumen-Diskretisierung betrachtet. Das lineare Mehrgitter erwies sich als signifikant stärkerer Vorkonditionierer als ein klassisches Jacobi iteratives Verfahren. Dies führte zu einer Reduktion der Laufzeit. Es wurde weiterhin gezeigt, dass die Möglichkeit für weitere Optimisierungen besteht, anhand der Umverteilung der relativen Last zwischen groben und feinen Leveln. Für den Transfer des Residuums zwischen den Leveln wurden ein volumengewichteter und uniform gewichteter Operator verglichen. Es zeigte sich, dass ein volumengewichteter Transferoperator keine Vorteile bietet.

List of Figures

List of Tables

List of Symbols and Abbreviations

Abbreviations

$C^2A^2S^2E$ Center for computer applications in aerospace science and engineering

BE Backwards Euler

CAS Center for computer applications in aerospace science and engineering

CFD Computational fluid dynamics

CFL Courant-Friedrichs-Lewy

CODA CFD for ONERA, DLR and Airbus

CoMMA Coarse mesh multigrid agglomerator

CPU Central processing unit

DG Discontinuous Galerkin

DLR German Aerospace Center

FMG Full multigrid

FV Finite volume

GMRES Generalized minimum residual method

GPU Graphics processing unit

HPC High performance computing

LMG Linear multigrid

MDA Multidisciplinary analysis

MDO Multidisciplinary optimization

NASA National Aeronautics and Space Administration

NMG Non-linear multigrid

ONERA Office national d'études et de recherches aérospatiales

PDE Partial differential equation

SER Switched evolution relaxation

Roman letters

$\bar{J}$ Linearized jacobian

$\bar{M}$ Mass matrix

R Residual

u State vector

F^c Convective flux

F^v Viscous flux

E Total energy

Greek letters

ν_t Eddy viscosity

ρ Density

Contents

Chapter 1

Introduction

Higher-Order methods for computational fluid dynamics (CFD) have gained significant traction in recent years. The discontinuous Galerkin method, originally developed for the discretization of the neutron transport equation [12], discretizes the partial differential equation (PDE) using element-wise local polynomials. Due to the resulting discontinuities at element interfaces, the method is a very natural choice for hyperbolic PDEs. Higher-order discretizations can be achieved by increasing the maximum polynomial degree of the polynomials. This results in a stiffer numerical system due to increased degrees of freedom and bigger off-diagonal contributions. As such, efficient numerical methods for the non-linear solution process of steady-state problems need to be developed for use in industrial or industry-relevant problems. A method to accelerate and stabilise the solution process is the non-linear multigrid, which will be described and characterized in this report.

In the first section of this report, the relevant theoretical background necessary for following the report will be laid out. Afterwards, the performed experiments will be introduced and their results analysed. Furthermore, the results of a preliminary analysis of a linear multigrid method as a preconditioner will be presented. In the last two sections, the work done and the results will be summarised and recommendations for future work will be given.

Chapter 2

Theoretical Background

This section will introduce the relevant concepts needed for the execution of the research project. In the beginning, a brief explanation of the relevant equation and its discretization using a discontinuous Galerkin method will be given. Afterwards, the standard non-linear solution process will be explained. In the following subsection, different multigrid methods are presented, which form the basis of the conducted research. Subsection 2.4 describes a method, which enables DG discretizations on agglomerated meshes. The section concludes with an explanation of the chosen transfer operators as well as their implementation.

2.1. Discontinuous Galerkin Discretization of the Reynolds averaged Navier Stokes Equations

The steady Reynolds averaged Navier Stokes equations with the Spalart-Allmaras eddy viscosity model (RANS-SA) in its negative formulation[13, 1] can be expressed in terms of its convective fluxes $\mathcal{F}^c$ and viscous fluxes $\mathcal{F}^v$, with source terms $\mathcal{S}$ as

$$\nabla \cdot \left(\mathcal{F}^c\left(\mathbf{u}\right) - \mathcal{F}^v\left(\mathbf{u}, \nabla\mathbf{u}\right)\right) - \mathcal{S}\left(\mathbf{u}, \nabla\mathbf{u}\right) = 0 \tag{2.1}$$

with the state vector $\mathbf{u} = \left[\rho, \rho u, \rho v, \rho w, \rho E, \rho\tilde{\nu_t}\right]^T$. The solution is expressed in terms of element local polynomials. The basis for the space of ansatz and test functions is defined in physcial space as a modal basis of orthonormal functions. This basis spans the same space as a Taylor basis [10] while having a lower condition number of the element mass matrix [5]. A more detailed description of the choice of basis can be found in section 2.4

Due to the discontinuity at the element interfaces, the discontinuous Galerkin method is a very natural choice for hyperbolic equations as shown in [3]. The convective fluxes are calculated using Roe's method with an entropy fix of 0.1, proposed by Harten and Hyman [7]. For the treatment of viscous terms, the second scheme of Bassi and Rebay (BR2) [6] is used. In the present code, there is the possibility for using the first scheme of Bassi and Rebay (BR1) [2], however no full Jacobian of this scheme is present and thus the BR2 is used in all cases. The equations are solved in a fully coupled manner, thus the full Jacobian has to be calculated.

Integration on the elements or faces is performed using a tensor product Gauss quadrature.

2.2. Non-linear solution process

In order to arrive at a steady-state solution, the problem is iterated in pseudo time using a linearized backward Euler method, which is the standard in the present solver [9], defined in 2.2 for the time step i, where $\bar{M}$ is the mass matrix of the system which for an orthonormal basis will be the identity matrix. $\bar{J}$ represents the linearized Jacobian of the Residual $\mathbf{R}$

$$\left[\frac{\bar{M}}{\alpha_i \Delta t} + \bar{J}\right]\left(\mathbf{u}_i - \mathbf{u}_{i-1}\right) = \mathbf{R}_i \tag{2.2}$$

The local pseudo timestep $\alpha_i \Delta t$ is chosen, such that with decreasing residual, the timestep factor α_i will increase. This will recover a Newton algorithm for $\lim \alpha_i \to \infty$. In practice, a cell local timestep

is used which is specified by the Courant-Friedrichs-Lewy (CFL) number. The resulting linear system can be solved to obtain a correction for $\mathbf{u}$. The solution process of the system does not have to be exact, as no time accurate time stepping is needed and only an improvement of the non-linear residual is requested. The system is solved using a preconditioned GMRES. The preconditioning of the matrix is done using a line inversion and a fixed number of Jacobi iterations using the Spliss library [15].

Switched evolution relaxation (SER) is a formulation that increases the pseudo time step size according to the reduction of the residual [11]. The CFL ramping is controlled mainly by two parameters, the initial CFL number and the SER exponent, from which the CFL number can be calculated as shown in equation 2.3, where n is the SER exponent and R_i is some measure of the residual, which is compared to a reference Residual R_0. As no time-accurate solution is sought, the pseudo timestep can be calculated cell local from the CFL number.

$$CFL_i = CFL_{initial} / \left(R_i / R_0\right)^n \tag{2.3}$$

The used code implements 2 versions of the Residual norm R, where either the smallest or the biggest residual of the transport equations is dominating R, yielding in the first case more aggressive and in the latter case a more stable CFL ramping. For all experiments performed, the latter case was chosen.

2.3. Multigrid methods

Multigrid methods were devised to accelerate the convergence of numerical simulations. These methods use a sequence of coarser discretizations, on which a modified version of the original problem is solved. The solution of the coarse discretizations is then used to correct the initial guess of the fine level problem. Due to the reduced degrees of freedom on the coarse levels, higher pseudo time steps can be employed on these levels [14]. This section will lay out the general multigrid methods relevant for this work, starting from a linear multigrid and moving on to multigrid methods for non-linear problems. Finally, nested multigrid methods for DG methods are introduced.

Coarse level for Discontinuous Galerkin methods In the context of DG methods, two different types of coarsening can be applied for generating coarse levels. The first type of coarsening is similar to the approach taken in Finite Volume methods, such as described in [8], where the problem is discretized on an agglomerated coarse mesh. As in the case of DG methods, the solution on an element is described by basis functions up to a certain order, a coarse level can alternatively be formed by restricting the space of the basis functions for each element.

Linear multigrid The linear multigrid (LMG) is the simplest multigrid algorithm, and due to the linearity of the algorithm and underlying problem also very well analyzed[14]. The general algorithm for the solution of $\bar{L}\mathbf{u} = \mathbf{f}$ is detailed in algorithm 1, adapted from [16], where $\bar{L}$ is a discrete linear operator, in the form of a matrix, acting on the state vector $\mathbf{u}$. On each level, the smoothing operation S is performed, which can be for example a certain number of Jacobi iterations, as described in [16]. The transfer of states between different levels is done using the transfer operator I_b^a, which transfers a state from level b to level a. The construction and properties of these operators are discussed in a later section in detail. $\bar{L}^l$ is the linear operator on the coarse level, which can either be constructed through a rediscretization of the problem or using a Galerkin transfer of the fine level operator in the form of equation 2.4.

$$\bar{L}^{l-1} = I_l^{l-1} \bar{L}^l I_{l-1}^l \tag{2.4}$$

The linear multigrid algorithm is implemented in Spliss and can be used as a pre-conditioner for the GMRES algorithm. The linear solver is required in the non-linear solution process as part of the linearized backwards Euler method as described in section 2.2.

Non-linear Multigrid When solving a non-linear problem of the form $\mathcal{L}(u) = f$, the non-linearity of the operator $\mathcal{L}$ has to be taken into account. The general algorithm for a V-Cycle non-linear multigrid (NMG) is shown in algorithm 2, adapted from [16]. It should be noted that even though a complete solution of the coarse problem $Solve\left(\mathcal{L}^l(u) = f\right)$ is shown in the algorithm, this is not necessary and

Algorithm 1 Linear Multigrid $LMG\left(\mathbf{u}, \mathbf{f}, l\right)$

```
if l = 0 then
    u_n ← (L̄^l)^{-1} f
    return u_n − u
else
    for i = 1 to n_1 do
        u ← S(L̄^l, u, f)                                  ▷ Pre-smoothing
    end for
    u_{l−1} ← 0
    f_{l−1} ← I_l^{l−1} (f − L̄^l u)
    u ← u + I_{l−1}^l LMG(u_{l−1}, f_{l−1}, l − 1)
    for i = 1 to n_2 do
        u ← S(L̄^l, u, f)                                  ▷ Post-smoothing
    end for
end if
```

can be replaced by an approximate solution of the problem. As to solve or smooth the non-linear problem on each level a linearized backwards Euler method is employed it is important to choose a suitable timestep on each level. Theory from the linear case indicates that this timestep can be higher than the timestep on the finest level [14]. As described in section 2.2, the pseudo time step at each iteration is chosen according to the residual of the problem compared to some initial residual. Using the freestream residual as a reference on each coarse level results in a time step choice that will approach the optimal convergence of a Newton iteration independently of the other levels.

Algorithm 2 Non-linear Multigrid $NMG\left(u, f, l\right)$

```
if l = 0 then
    u_n ← Solve(L^l(u) = f)
    return u_n − u
else
    for i = 1 to n_1 do
        u ← S(L^l, u, f)                                  ▷ Pre-smoothing
    end for
    u_{l−1} ← I_l^{l−1} u
    f_{l−1} ← f_{l−1} + I_l^{l−1} (f − L^l(u)) − (f_{l−1} − L^{l−1}(u_{l−1}))
    u ← u + I_{l−1}^l NMG(u_{l−1}, f_{l−1}, l − 1)
    for i = 1 to n_2 do
        u ← S(L^l, u, f)                                  ▷ Post-smoothing
    end for
end if
```

Nesting of Non-linear Multigrids In order to convert a single grid algorithm into a simple multigrid algorithm, the smoother on the finest level, e.g. a linearized backwards Euler method is replaced by a non-linear multigrid. In a similar fashion, the smoother employed on the coarse levels can again be replaced by a non-linear multigrid, yielding a nested multigrid. In these methods, there exists an outer multigrid, which on its coarse levels, employs an inner multigrid method to smooth the error components. As there are two different types of coarsening in the DG case, it is chosen, that the inner multigrid will always be of a different type than the outer multigrid.

Nomenclature of multigrid methods In order to refer to different multigrid types and cycles for the non-linear multigrid, the multigrids are labelled by the total number of levels and their coarsening type. For example, a 3H multigrid is a non-linear multigrid with 2 coarse levels, which are constructed using an agglomeration based strategy. For nested multigrid methods, the outer multigrid will be named first afterwards the inner multigrid on the coarse level will be named. For example, a 2P-3H multigrid

is a nested multigrid with 1 coarse level with p-coarsening, which will employ a 3H multigrid as a smoother.

2.4. Discontinuous Galerkin discretization on agglomerated meshes

The agglomeration of the mesh is done by the external library CoMMA, developed by ONERA. CoMMA is a graph-based agglomerator. An important feature of the algorithm is the available semi coarsening, which aims to agglomerate cells in the boundary layer in the wall-normal direction, to increase the isotropy of the mesh. With the standard settings, 4 cells are agglomerated together in the isotropic regions of a 2-dimensional mesh, or 8 cells are agglomerated for 3-dimensional meshes. The following section will provide a description of how DG methods can be used on such agglomerated meshes.

All meshes considered in this report are conformal meshes consisting of elements $\{T_i\}$. A coarse mesh is formed such that each fine element T is assigned to exactly one coarse element $\mathcal{T}$.

Geometry preservation The elements of the agglomerated mesh are handled, such that for the agglomerated element $\mathcal{T} = \{T_i\}$ all sub-elements T_i are kept. The boundary of $\mathcal{T}$ consists of several faces $\mathcal{F}$ where a face is defined as the interface between two agglomerated elements. A face can be decomposed such that $\mathcal{F} = \{F_i\}$, where F_i represents the facets belonging to the fine elements.

For performing integration on these agglomerated meshes, the direct sum of the integrals of the fine elements T_i is taken. For the integration of the faces of the mesh, the direct sum of the fine facet integrals is used analogously. This has previously been explored by [5] and [4], showing the increased accuracy of integration and keeping the degree of integration that was used on the fine mesh.

Orthonormal modal basis on agglomerated meshes In order to perform DG simulations on an agglomerated mesh, a basis for $\mathcal{P}^p$, the space of functions in $\mathbb{R}^3$ covered by polynomials up to a total degree p, on the agglomerated element $\mathcal{T} = \{T\}$ must be defined. The basis $\phi^p_{\mathcal{T}} = \left\{\phi^i_{\mathcal{T}}\right\}$ should satisfy the following property

$$\left\langle \phi^i_{\mathcal{T}}, \phi^j_{\mathcal{T}} \right\rangle_{\mathcal{T}} = \delta_{ij} \tag{2.5}$$

where $\langle a, b\rangle_{\mathcal{T}}$ is the inner product of a and b over the domain $\mathcal{T}$ [5]. As there exist no general reference element for a general agglomerated element on unstructured meshes, the basis functions must be calculated in physical space [16]. This is done by first defining a non orthonormal basis on $\mathcal{T}$, which is then orthonormalized using the modified Gram-Schmidt process. This leads to a well conditioned [5] basis on the agglomerated element. In practice this approach yields less degrees of freedom compared to a nodal base of similar order for all element types that are not tetrahedrons.

Coarse grid Jacobian In order to apply the linearized backward Euler smoother on the coarse multigrid level, the Jacobian of the coarse grid residuals with respect to the coarse grid state has to be formed. For this the problem can either be rediscretized on the coarse mesh or a Galerkin transfer of the matrix can be performed. The Galerkin transfer described in equation 2.6 uses the prolongation and restriction operators described in the following section to construct the coarse level matrix. In this report, a rediscretization is used for calculating the Jacobian in the case of non-linear multigrid. For the linear case, a Galerkin transfer was used.

$$\bar{J}^{l-1} = I^{l-1}_{l} \bar{J}^{l} I^{l}_{l-1} \tag{2.6}$$

2.5. Transfer operators

Transfer Operators are needed to restrict or prolongate the state between grid levels. These grids can vary in agglomeration degree as well as polynomial degree in the basis functions. For both operations, the condition stated in equation 2.7 has to be satisfied for the relation between the coarse grid state $\mathbf{u_{l-1}}$ and the fine grid state $\mathbf{u_l}$, which corresponds to a L_2 projection onto the coarse element basis functions.

$$\int_{\mathcal{T}} \boldsymbol{\phi}^i_{l-1} \left(\mathbf{u_l} - \mathbf{u_{l-1}}\right) dV = 0 \qquad \forall \boldsymbol{\phi}^i_{l-1} \in \boldsymbol{\Phi}_{l-1} \tag{2.7}$$

Where $\mathbf{\Phi}_{l-1}$ is the space of basis functions for a coarse element. In the case of orthonormal basis functions, the transfer operators for restriction and prolongation are the transpose of each other. In the case of only a change in polynomial order between grids, the transfer operator reduces to a truncated identity matrix or an identity matrix with extra zero rows, which again are the transpose of each other.

2.6. Implementation

The transfer operators needed were implemented into the existing CFD-Code CODA. Algorithm 2 was already present in the code for Finite Volume discretizations and was extended to DG as well as adapted to allow nesting of multigrids as described in section 2.3. An important side effect of the implementation of the multigrid agglomeration into the existing code is a simplification of the widely used p-startup, which is used to initialize higher-order simulations from a lower order precursor simulation. This was previously done externally and requires an explicit transfer of the solution, while the multigrid implementation provides a natural way for p as well as h-sequencing. Furthermore, due to the nesting property of the implementation, a combination of p and h-sequencing can be used for simulations on very numerically challenging meshes.

For the integration of the linear multigrid, work in CODA, as well as the linear solver library Spliss, had to be done. An implementation of the linear multigrid algorithm existed in Spliss, to be used with a uniform residual restriction, however, an extension was implemented to allow the use of general linear transfer operators using a Galerkin transfer. Furthermore, the linear solver interface on the CODA side was extended to allow the use of the linear multigrid as a preconditioner, for which the mesh agglomeration was used.

Chapter 3

Numerical Experiments

The following section will explain the numerical experiments conducted to assess the performance of the previously described multigrid methods. The test cases are ordered in ascending order of complexity starting from inviscid compressible flow over a smooth bump to finishing with a computation of a multi-element airfoil in high lift configuration.

3.1. Verification

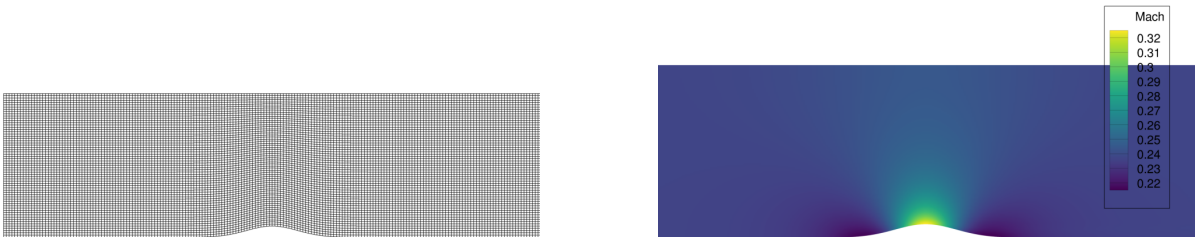
As this numerical method is novel in the code, care has to be taken to confirm the correctness of results obtained with this method. The criterion for convergence used is a reduction of the initial density residual or turbulent viscosity residual, depending on the test case, by a factor 10^{-11}. This reduction is evaluated on the finest level of the multigrid, thus it is equivalent to determining convergence of a single grid method, assuming uniqueness of the solution. Consistency for the single grid has been verified previously by the developers as well as by external partners. Due to this reason, no additional verification of consistency is performed.

3.2. Inviscid compressible flow over a smooth bump

This 2-dimensional test case computes the subsonic Euler equations over a smooth Gaussian bump. The mesh consists of 12288 quadratic hexahedrons and can be seen in figure 3.1, accompanied by the Mach number distribution of the converged solution. The top and bottom walls are slip walls. The inflow is prescribed by a fixed total pressure and total temperature, with the outflow being fixed to a static pressure as well. This test case has been present in all international Higher-Order CFD workshops as a verification test case. To assess the robustness of the different methods with respect to the pseudo-time-stepping, a sweep of initial CFL numbers was performed.

Simulation Parameters The solution was computed using a 3rd-order Discontinuous Galerkin discretization, using 10 degrees of freedom per element and state variable. Of these 10 degrees of freedom, only 6 degrees are relevant for the computation as all basis functions with $\frac{\partial \phi_i}{\partial y} \neq 0$, have to be 0 by the nature of the present boundary conditions. The SER exponent was fixed to 0.6 with a global non-linear damping of the update fixed to 0.8. For the solution of the linear system, a Line Inversion Jacobi preconditioner was used in combination with a GMRES solver. The number of Jacobi iterations was fixed to 50 iterations. The solution process of the linear system was stopped when the linear residual was reduced by 6 orders of magnitude or after 20 Krylov iterations, whichever comes first. The solution was iterated till the density residual of the solution had been reduced by 11 orders of magnitude. The results of the simulations are shown in figure 3.2. If a simulation did not converge for a specific CFL number, the data point is not shown.

Results In figure 3.2 it can be seen, that all multigrid methods provide an increased stability envelope when compared to the single grid solution. This increase in stability is visible due to the increased extension of the lines into the right-hand side of the figure indicating tolerance to more aggressive CFL settings. It can be seen that the 3H and the 3p-multigrid perform better than the single grid solution for all CFL numbers. Furthermore, the 2p-multigrid performs better than the singlegrid solution for all



(a) (b)

Figure 3.1: (a) Quadratic mesh of the Gaussian smooth bump testcase with 12288 elements. (b) Mach number distribution of the converged solution.

initial CFL numbers bigger than 5. A slight difference in stability between the h- and p-multigrids can be observed.

When comparing these simple multigrid methods to nested multigrid methods, it can be seen that the nested methods provide an even bigger stability region than simple multigrid methods. Furthermore, a decrease in run time for all CFL numbers is observed. The influence of the initial CFL number on the total run time also diminishes for the considered nested multigrid methods. The jaggy nature of the curves for the nested multigrid arises from the fact that only very few iterations are needed for convergence. Thus a reduction in iteration number was sometimes overshadowed by an increased cost per iteration.

This test case was also used to validate the assumptions made in section 3.1. The result of the investigation can be found in table 3.1.

Table 3.1: Absolute values of C_L and C_D for the smooth bump test case with indication of relative difference compared to the single grid solution.

Algorithm	C_L	$\Delta C_L/C_L$	C_D	$\Delta C_D/C_D$
SG	1.2465316609E-04	0	9.6844799587E-08	0
2p	1.2465316564E-04	-3.6019943116E-09	9.6844838037E-08	3.9703112787E-07
3p	1.2465316607E-04	-1.5242286376E-10	9.6844685873E-08	-1.1741900492E-06
3h	1.2465316515E-04	-7.4927900448E-09	9.6844771377E-08	-2.9129080884E-07
2p-3h	1.2465316567E-04	-3.3853933542E-09	9.6844829401E-08	3.0785339146E-07
3p-3h	1.2465316533E-04	-6.0407610015E-09	9.6844796057E-08	-3.6452137937E-08
3h-2p	1.2465316549E-04	-4.7732440863E-09	9.6844850188E-08	5.2249888708E-07
3h-3p	1.2465316553E-04	-4.4603759884E-09	9.6844842224E-08	4.4026421843E-07

From table 3.1 it can be seen that the lift coefficient calculated with the multigrid methods match the single grid lift coefficient very well. The slightly higher discrepancy in the drag coefficient can be explained by the lower value of the reference coefficients. In conclusion, it can be seen that the absolute discrepancy lies on the order of 10^{-13} to 10^{-15} which is very close to machine epsilon and can thus be considered non-existent for all intends and purposes. This proves the validity of assumptions made in section 3.1 for this specific test case.

The analysis of multigrid methods for this test case show the potential of simple multigrid methods as well as the potential of novel nested multigrid methods. These results warrant further investigation into the behaviour of nested multigrid methods on more complex test cases.

3.3. Turbulent flat plate boundary layer

The 2-dimensional zero pressure gradient flat plate test case is a common verification test case for CFD solvers and a more detailed description can be found on the NASA turbulence modelling resource page[1]. In this test case, the RANS equations with the negative formulation of the Spallart Allmaras[1] turbulence model are solved. In contrast to the classical hexahedral mesh, a prism mesh was chosen, which is obtained by splitting each hexahedron into 2 prisms. Using this mesh a very stiff numerical problem is

[1] `https://turbmodels.larc.nasa.gov/flatplate.html`

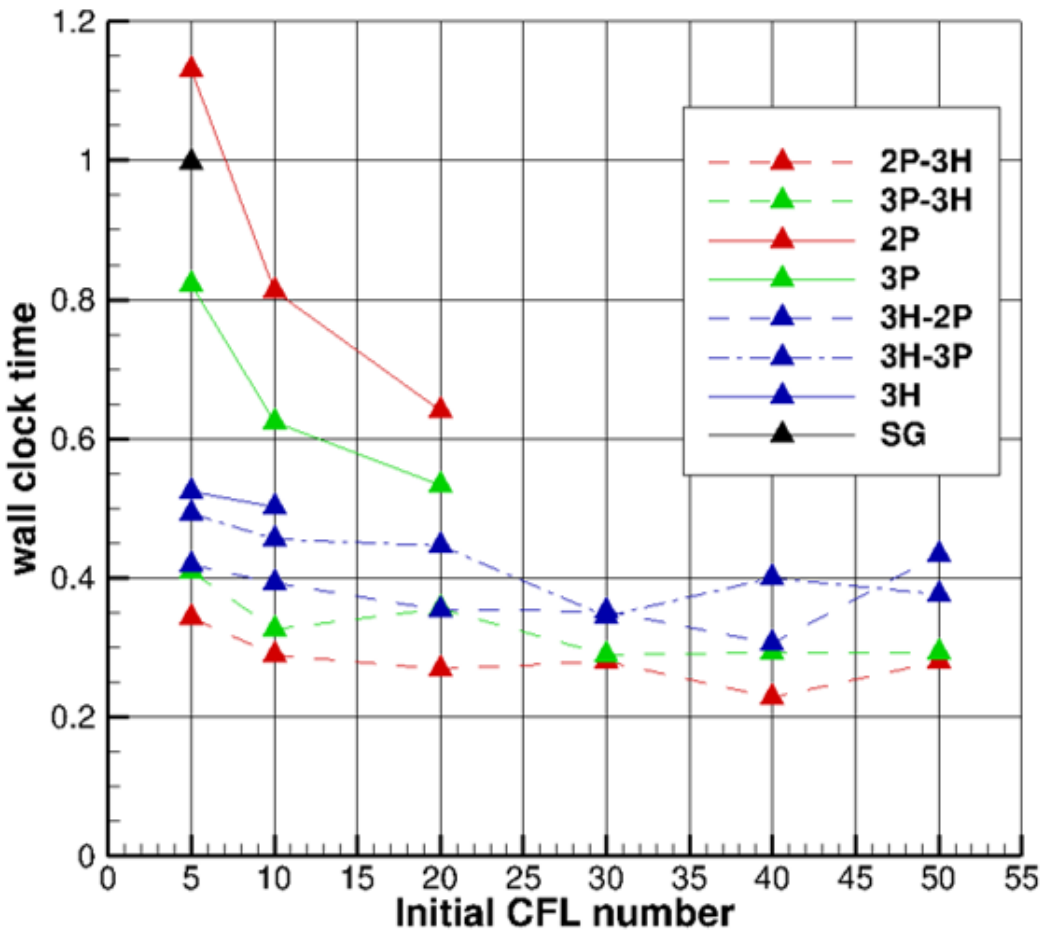


Figure 3.2: Normalized recorded wall-clock time for the smooth bump test case for different multigrid methods against initial CFL numbers. Dashed lines represent nested multigrid methods while continuous lines represent simple multigrid methods.

obtained. The stiffness of the numerical problem is evaluated by comparing the computational time and iteration number between a hexahedral and a prismatic mesh. These meshes are coarser than the mesh used for the final evaluation. For this, a nested startup procedure was used for the prismatic mesh to arrive at an initial condition, while the hexahedral mesh used a simple p-startup. Both computations were done as single grid simulations after the startup was complete. The comparison of computational time can be found in table 3.2. From this comparison, it can be seen, that the hexahedral mesh can

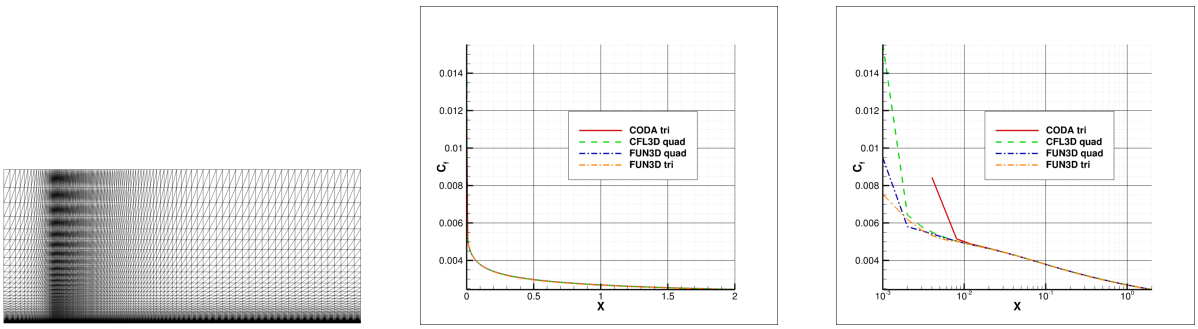
Table 3.2: Comparison of hexahedral and prismatic mesh for the 2 dimensional zero pressure gradient flat plate.

Mesh	No. of Elements	Initial CFL	normalized wct	Iterations
Hexahedral	13056	1000	1	5
Prismatic	6528	250	2.43	72

be run at a significantly higher CFL number indicating a less stiff numerical problem. Even though the prismatic mesh only uses half as many elements when compared to the hexahedral mesh it takes significantly longer to converge needing approximately 2.5 times the computational time and 14 times the number of iterations compared to the hexahedral mesh. From these preliminary results, it can be seen that a prismatic mesh for the flat plate case is a computationally very challenging test case compared to a more standard hexahedral mesh. Thus the prismatic case was chosen to investigate the behaviour of multigrid methods further.

The mesh used for the final analysis of this test case can be seen in figure 3.3(a) and consists of

26112 prisms. A comparison of the solution to a reference solution provided by NASA [2] can be seen in figure 3.3(b) and figure 3.3(c). The solution computed matches the reference solution very well, thus no influence of the non-linear solution process on the converged solution is present. The subsonic inlet is defined by its total pressure and total temperature, while the outlet is kept at a fixed static pressure. The upper boundary is treated by a farfield boundary corresponding to a reference state of Mach 0.2. The flat plate is enforced by a non-slip wall, which is located behind a symmetry condition that spans approximately the first 20 percent of the domain length. The flow is flowing with a Reynolds number of 5×10^6 in reference to a unit length.



(a) (b) (c)

Figure 3.3: (a) Mesh used for the 2 dimensional zero pressure gradient flat plate test case. (b/c) Comparison of the skin friction coefficient between a third order DG solution on the third finest prismatic mesh using CODA and reference solutions provided by NASA on the finest hexahedral and prismatic meshes.

Simulation setup A 3rd-order DG discretization is used for the spatial discretization and the solution is considered converged if the turbulent viscosity residual is reduced by 11 orders of magnitude on the finest mesh. As the numerical problem is very stiff, it is not possible to immediately start the computation as a third-order solution, even a commonly used p-startup is not effective enough to overcome this stiffness to arrive at a sufficiently good initial condition unless a nested multigrid algorithm is used. Thus a nested startup procedure is used where first a solution on the coarsest mesh is computed using a first-order DG discretization. Afterwards, the order of solution is increased to the final requested order and then the agglomeration level of the mesh is reduced until an initial condition for the final stage is found. The linear systems arising using a GMRES Krylov subspace solver preconditioned with 50 Jacobi iterations, which in turn are preconditioned using a Lines Inversion. The linear system is again solved till a reduction of the linear residual of 6 orders of magnitude is reached or till 20 Krylov iterations are reached, whichever comes earlier. For the h-multigrid cycles 2 coarse levels are used, while for the p-multigrid cycle, only a single coarse level is used. In unreported work, it was observed that a coarsening down to first order was not beneficial in terms of convergence and could even lead to divergence, due to the inconsistencies of viscous flux terms in the first order DG discretization compared to second and higher-order DG schemes. The differing parameters for the simulations can be found in table 3.3. The p-multigrid was not able to converge due to difficulties in the startup. The results for the remaining methods can be seen in figure 3.4.

Table 3.3: Time step settings used for the 2DZP flat plate case with a triangular mesh

multigrid method	SER exponent	initial CFL
3v h-NMG	0.4	150
2v p-NMG	-	-
2v3v ph-NMG	0.4	1000
sg	0.4	150

[2] `https://turbmodels.larc.nasa.gov/flatplate.html`

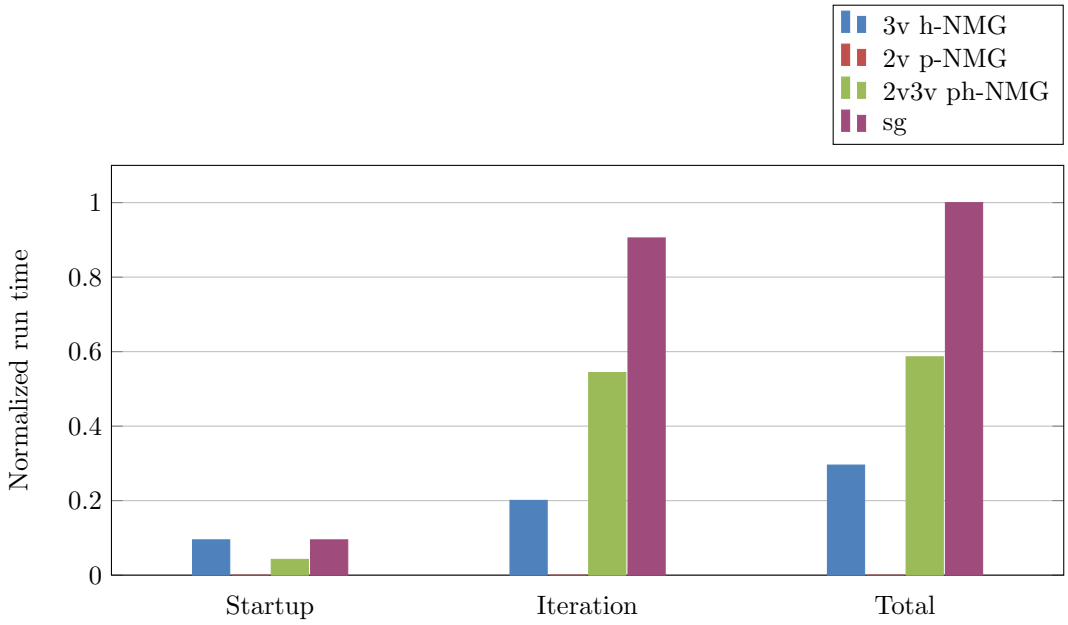


Figure 3.4: Run-time comparison between different multigrid algorithms broken down into startup and iteration time for the zero pressure gradient flat plate.

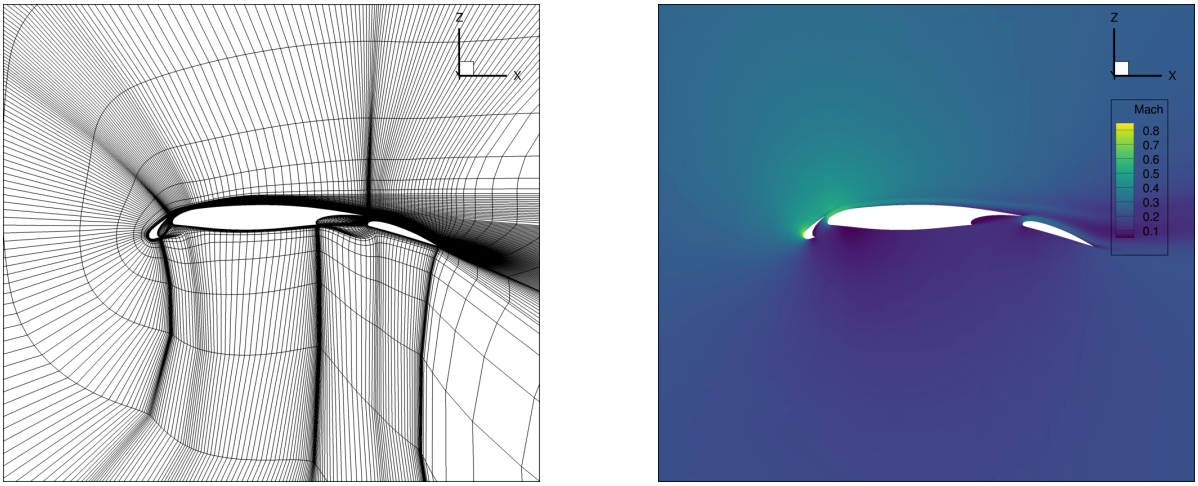
Results From the figure the advantage of agglomeration based multigrid methods for this test case can clearly be seen. However, the pure p-multigrid was not able to converge for this mesh. This can be explained by the fact, that the prisms in the mesh are agglomerated into hexahedral elements in the boundary layer, leading to a significantly easier to solve problem, a p multigrid can not exploit this geometric relationship. It can also be seen, that the h-multigrid is approximately twice as fast as the ph-multigrid, as the h-multigrid only performs 1 smoothing operation on the prismatic mesh, while the ph-multigrid performs 2 smoothing operations on the prismatic mesh at different orders. A significant advantage of the ph-multigrid is observed during the startup phase, where this method performs approximately twice as good as the h-multigrid and single grid startup, due to the fact, that each iteration could already use a multigrid method due to the nested properties similar to a full multigrid (FMG). In comparison to the single grid, both methods perform significantly better with a speedup of 1.7 and 3 for the ph- and h-multigrid respectively.

Even though the h-multigrid is by far the fastest method, the ph-multigrid was able to run at a significantly higher CFL number, compare to table 3.3. This indicates the increase in stability of nested multigrid methods, which was already observed in the previous section. It should be noted, that the advantages of agglomeration based h-multigrid methods only seem to emerge for cases with a sufficiently high number of elements.

3.4. L1T2 high lift airfoil

The L1T2 high lift multi-element airfoil is used to test the implemented methods on a more praxis relevant test case. For this first a structured mesh with 17972 quadratic hexahedrons is used, which can be seen in figure 3.5, accompanied by the Mach number distribution of the converged solution. The flow is simulated for a chord Reynolds number of 3.52×10^6 at Mach 0.197. The airfoil is kept at an angle of attack of 20.18 degrees.

Simulation Setup Similarly to previous cases, a 3rd-order DG solution is sought and considered converged as soon as the density residual has reduced by 11 orders of magnitude when compared to the freestream residual. To initialise the simulation, a p sequencing is used in all cases, such that the starting point of the 3rd stage is a fully converged second-order DG solution. The arising linear problem is solved using a GMRES which is preconditioned using a Lines Inversion and 100 Jacobi iterations. The linear system is considered converged after a linear residual reduction by 6 orders of magnitude or 20 Krylov iterations, whichever comes earlier. The initial CFL numbers and SER exponents for the



(a) (b)

Figure 3.5: (a) Near field of the structured mesh used for the L1T2 high lift multi element airfoil, with 17072 quadratic hexahedrons. (b) Mach number distribution of the converged solution.

different multigrid techniques can be seen in table 3.4. The relative run time of the cases can be seen in figure 3.6 and the required number of iterations on the finest level are shown in table 3.5 .

Table 3.4: Time step settings used for the L1T2 test case

multigrid method	SER exponent	initial CFL
3v h-NMG	0.6	48
2v p-NMG	0.6	48
2v3v ph-NMG	0.6	112
sg	0.6	36

Table 3.5: Number of smoothing steps on the finest level needed to convergence for the L1T2 airfoil.

multigrid method	smoothing steps
3v h-NMG	9
2v p-NMG	15
2v3v ph-NMG	11
sg	18

Results Taking a look at the time step settings used for the test case shown in table 3.4 it can be seen that again the multigrid methods provide an improvement in robustness in terms of initial CFL number when compared to the single grid solution. This increase in stability is even more pronounced in the case of the nested multigrid method, which can be run at a more than three times higher CFL number than the single grid case. In figure 3.6, the relative time taken of each method can be seen for both stages of the computation. It can be seen that all methods bar the h-multigrid perform very similarly in terms of total time. The h-multigrid takes approximately 25% longer than the other methods while taking only half the number of smoothing steps on the finest level as can be seen in table 3.5. This confirms that the multigrid implementation is correct, however points to performance issues of the current implementation.

3.5. CRM-HL wing section

This test case is very similar to the previous test case in terms of expected flow phenomena and general geometry. The airfoil is a wing section cut of the Common Research Model in High Lift configuration.

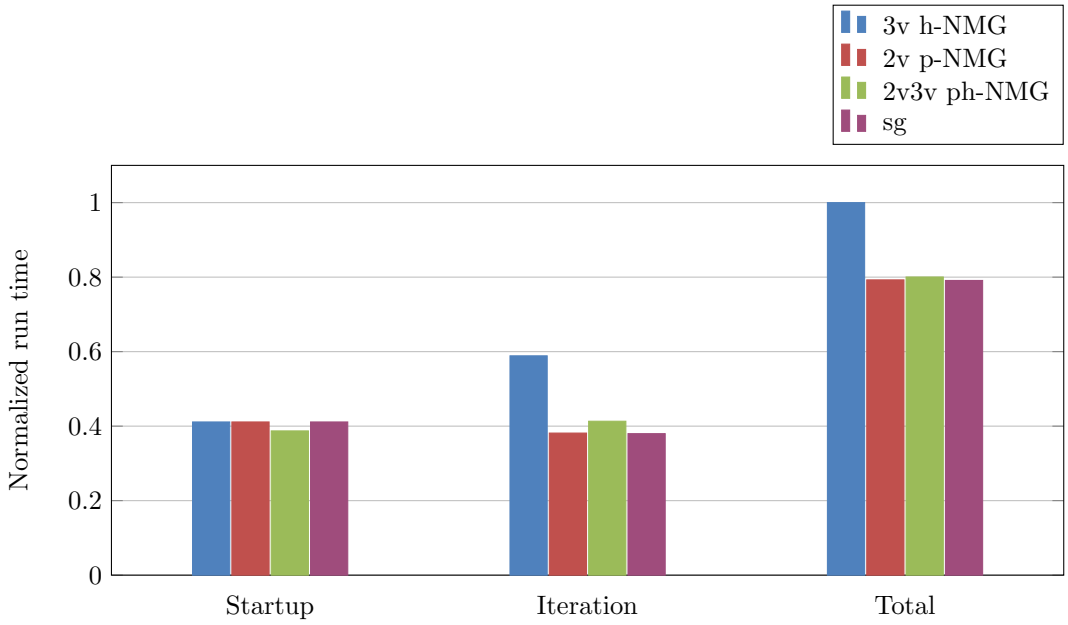
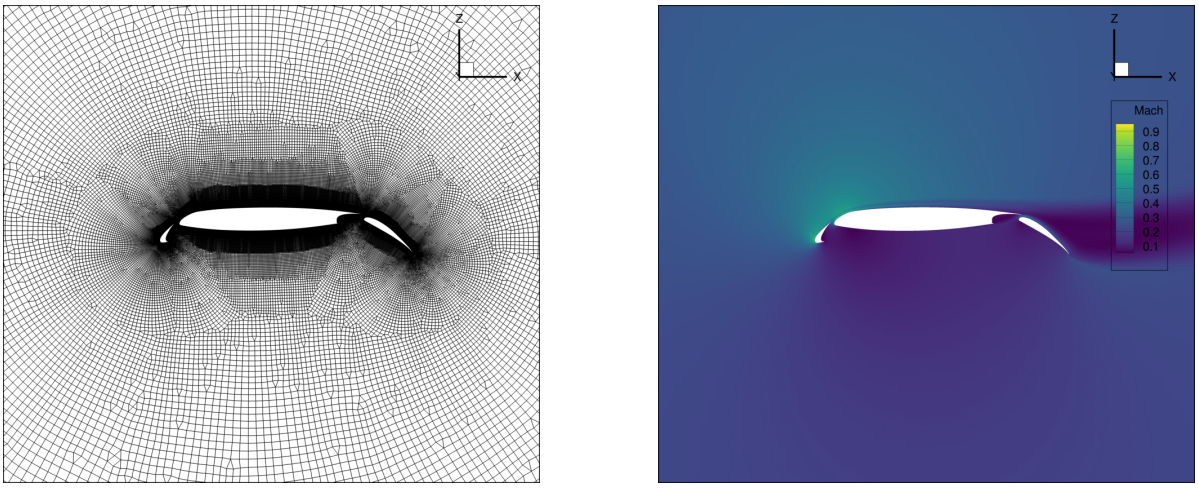


Figure 3.6: Run-time comparison between different multigrid algorithms broken down into startup and iteration time for the L1T2 multi element airfoil.

This mesh is chosen due to its hybrid nature including both prisms and hexahedrons, which provides a more numerically challenging mesh compared to the previous structured mesh. The mesh consists of 174443 cells. The flow state is the same as the previous case with a chord Reynolds number of 3.52×10^6 at Mach 0.197. The airfoil is kept at an angle of attack of 20.18 deg. The mesh can be seen in figure 3.7, together with the Mach number distribution of the near field for the converged solution.



(a) (b)

Figure 3.7: (a) Near-field mesh of the Common Research Model High-lift wing section with 174443 elements. (b) Mach number distribution of the converged solution.

Simulation Setup A 3rd-order DG solution is computed. For this, a 3 stage computation is performed in which the order of the discretisation is increased after every stage. Each stage is considered converged when the density residual has been reduced by 11 orders of magnitude compared to the freestream residual. The linear system at each pseudo time step is solved using a GMRES solver preconditioned with 100 Jacobi iterations and using a Lines Inversion. The iterative solution process of

the linear system is stopped when the residual is reduced by 6 orders of magnitude or when 20 Krylov iterations have been performed. The initial CFL numbers and SER exponents for the SER ramping are reported in table 3.6. The relative computational time spent can be found in figure 3.8 and the number of required smoothing steps can be seen in table 3.7.

Table 3.6: Time step settings used for the CRM-HL airfoil test case

multigrid method	SER exponent	initial CFL
3v h-NMG	0.8	48
2v p-NMG	0.8	48
2v3v ph-NMG	0.8	64
sg	0.8	48

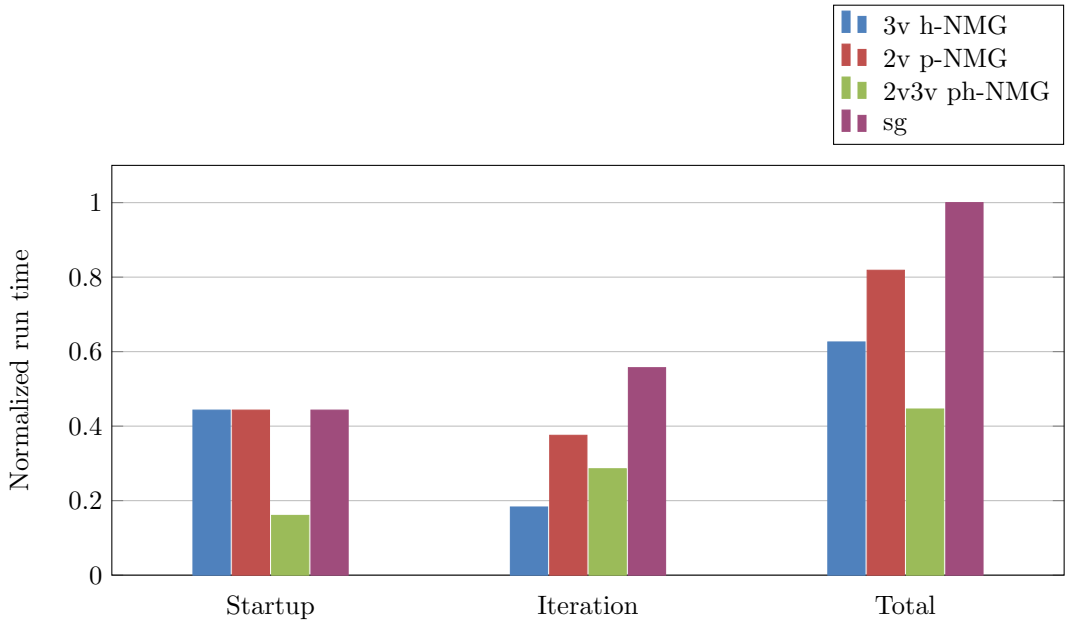


Figure 3.8: Run-time comparison between different multigrid algorithms broken down into startup and iteration time for the CRM-HL multi element airfoil.

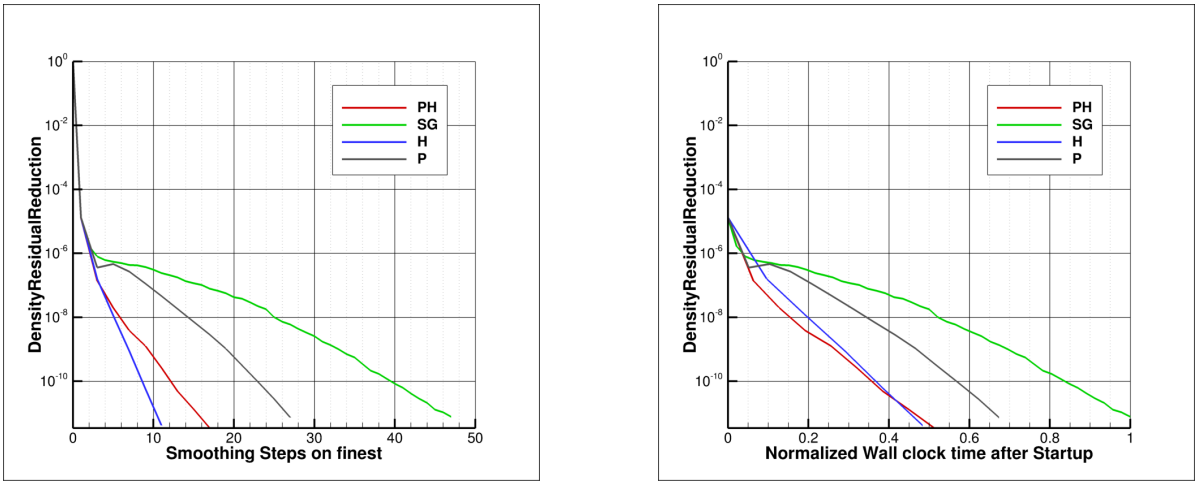
Table 3.7: Number of smoothing steps on the finest level needed to convergence for the CRM-HL airfoil.

multigrid method	smoothing steps
3v h-NMG	11
2v p-NMG	27
2v3v ph-NMG	17
sg	47

Results From figure 3.8 it can be seen that all three considered multigrid methods outperform the single grid calculation. In the total runtime, the ph-NMG performs the best of all methods providing a speedup of a factor over 2. In terms of time spent iterating the finest stage of the solution, the h-NMG performs the best, the ph-NMG, however, takes only half as long as other methods in the startup computation as due to the nested nature, an FMG like startup can be performed. Furthermore, the nested multigrid method is the most robust scheme in terms of initial CFL number as can be seen in table 3.6. In figure 3.9 the convergence of the density residual is plotted against the smoothing operations on the finest level as well as against the computational time after the startup computation is completed. It can be seen in figure 3.9(a) that the h-multigrid performs the best with respect to needed

smoothing steps. The nested multigrid combination of p- and h-multigrids performs somewhere between the h- and p-multigrid, while all methods outperform the single grid computation. When taking a look at figure 3.9(b), it is apparent that the difference in computational time between h and ph multigrid can partly be attributed to the fact that the ph-multigrid converges further below the requested residual as no partial non-linear iterations can be performed. It should be noted, that the p multigrid methods may be at an inherent disadvantage compared to h multigrid methods, as the simulation on the finest level is started from a fully converged second-order solution and thus the corrections of the p-multigrid can be expected to be smaller.

As the number of iterations needed for convergence is quite low, for all considered test cases, the performance of the linear solver in the linearized backwards Euler method has a significant impact on the performance of the whole method. As the linear systems arising from DG methods are quite stiff, due to the high number of degrees of freedom per element and variable, the linear solution process can take a significant amount of time and depends significantly on the strength of the used preconditioner. The use of a linear multigrid as a preconditioner will be explored in the next section.



(a) (b)

Figure 3.9: (a) Density resiudal of the solution plotted against smoothing steps on the finest level. The initial steep drop for the first step corresponds to the performed p-startup. (b) Density residual of the solution plotted against the wall-clock time after the startup is completed to compensate for the difference in startup computational time.

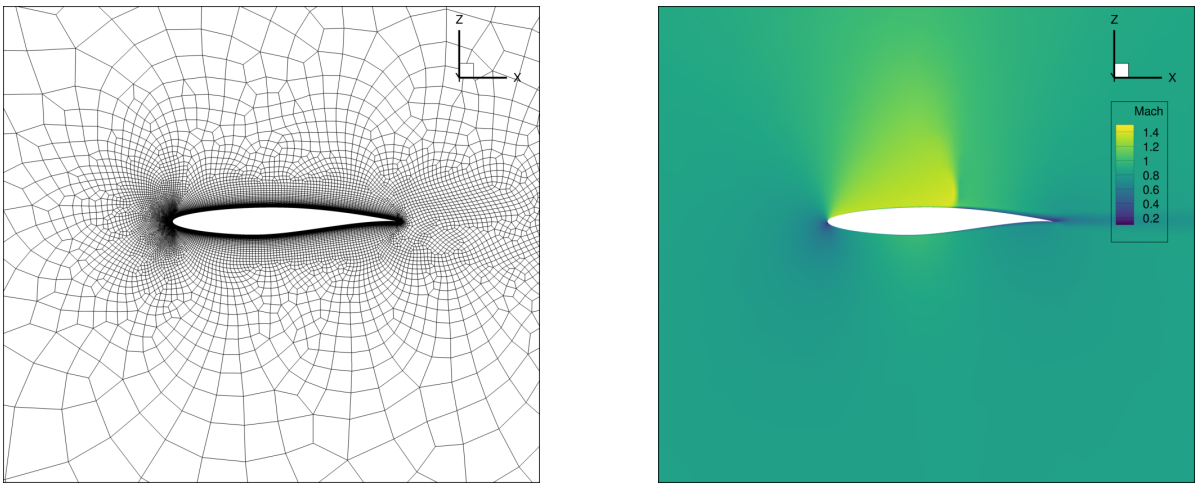
Chapter 4

Preliminary assessment of a linear multigrid preconditioner

This section will describe and analyse the results of the numerical experiments performed to assess the performance of the linear multigrid method as a preconditioner. This is done on the example of a Finite Volume (FV) computation of a transonic airfoil.

For the test case, the RAE-2822 airfoil was chosen with transonic flow conditions. This problem provides a frequently used and well-understood test case. Relevant to the agglomeration based linear multigrid is the presence of a shock on the suction side of the airfoil, as it can not be assured that no agglomeration of cells happens orthogonal to the shock, and the behaviour of the linear multigrid for such a situation can be assessed. For the simulation, an unstructured mesh with 25906 elements was chosen, this unstructured mesh provides a numerical challenge to the linear solver as the resulting system was found to be stiffer than for the structured case. The flow is simulated for a Reynolds number of 6.5×10^6 and a freestream Mach number of 0.734. The airfoil is kept at an angle of attack of 2.79 deg. A representation of the mesh and the flow solution of a second-order FV discretisation can be seen in figure 4.1.

The next section will first describe the general parameters used for the simulations. Afterwards, the influence of the chosen residual restrictor will be discussed. Finally, the influence of the relative work distribution between the coarse level and fine level will be assessed on the example of varying the work done on the finest level.



(a) (b)

Figure 4.1: (a) Near-field mesh of the unstructured RAE-2822 mesh with 25906 elements. (b) Mach number distribution around the airfoil for the described testcase.

4.1. Solver Parameters

The solution is computed using a second-order Finite Volume discretisation with gradient limiters. The gradient is reconstructed using a least-squares approach according to a parameter settings in a standard test case configuration. The non-linear system is solved using the previously mentioned linearized backwards Euler method employing cell local time steps, which are calculated using an SER ramping with an exponent of 0.4 and an initial CFL number of 10. The arising linear system is solved by using a preconditioned GMRES and the linear system is considered solved as soon as the linear residual is reduced by an order of magnitude or after 20 Krylov iterations, whichever comes earlier. For the preconditioning, the linear multigrid is compared to a Jacobi iterative solver.

The linear multigrid uses a single V-cycle and employs a smoother on every level. This smoother is a Jacobi iterative solver, which is preconditioned with an element local LU decomposition. The exact number of Jacobi iterations of the fine and coarse level smoother depends on the analysis performed and will be stated in the relevant sections. The non-linear problem is considered converged if the density residual has been reduced by 10 orders of magnitude compared to the freestream residual. The

4.2. Influence of fine level smoothing

First, the influence of the relative work distribution between coarse and fine level for the linear multigrid is evaluated. For this, a Jacobi iterative solver with 20 iterations and an element local LU decomposition preconditioner is used as the smoother on all coarse level. The iterations of the Jacobi iterative solver used as the fine level smoother is varied. This preconditioner is compared to a classic Jacobi preconditioner with an equivalent number of iterations. The convergence histories of the considered variations can be seen in figure 4.2.

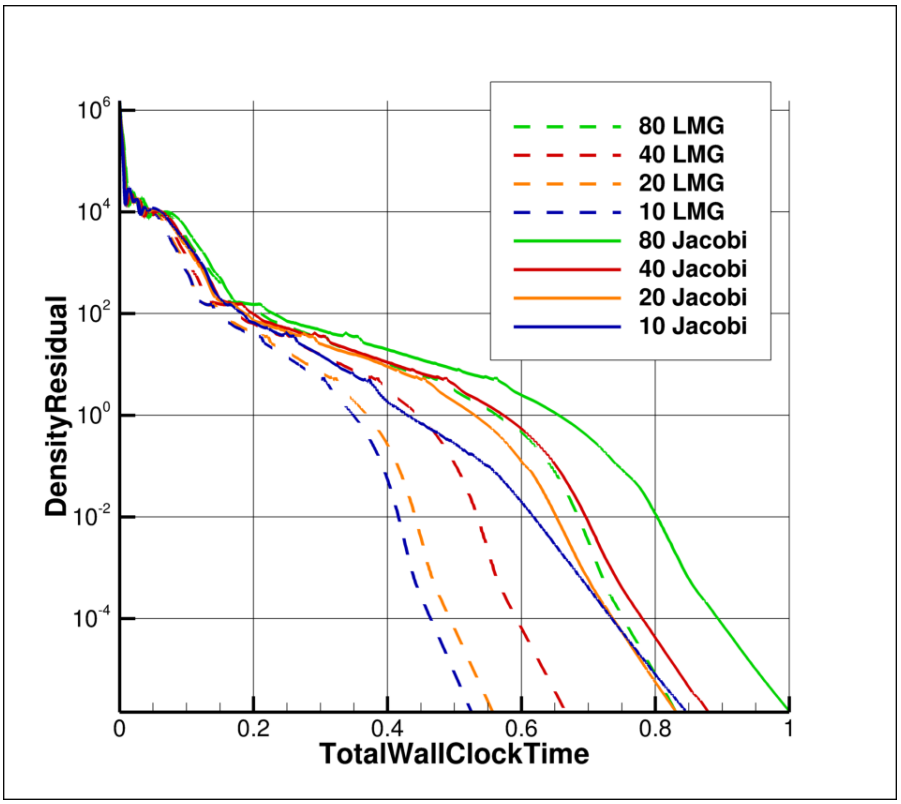


Figure 4.2: Convergence histories of different preconditioner combinations. Dashed lines indicate the use of a linear multigrid preconditioner, while continuous lines indicate the use of a Jacobi preconditioner. The numbers indicate the number of Jacobi iterations performed on the finest level. Lines of the same colour indicate the same number of Jacobi iterations on the finest level.

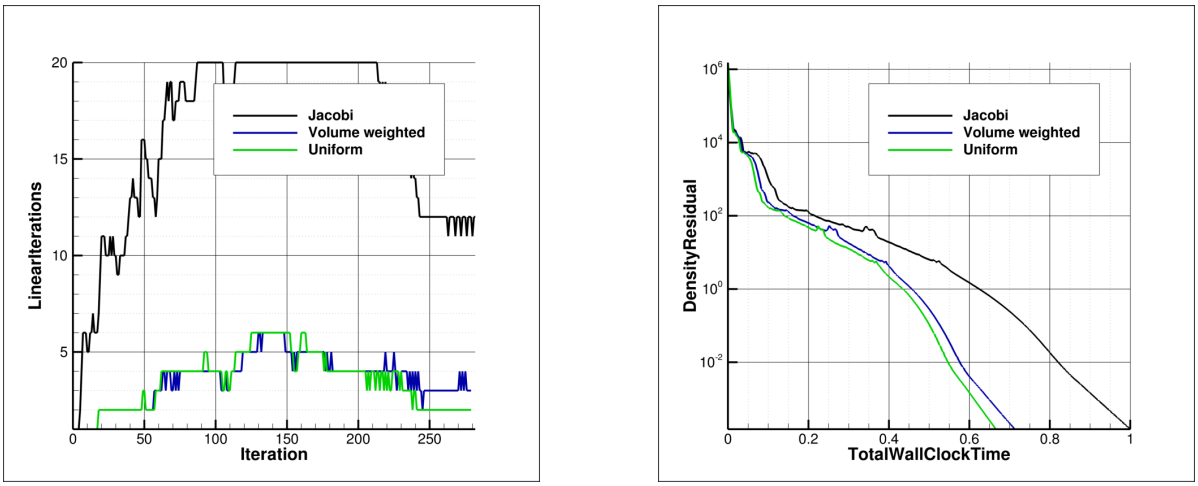
From figure 4.2 it can be seen, that a linear multigrid preconditioner always performs better than a Jacobi preconditioner with an equivalent number of iterations on the finest level. Furthermore, it is

observed, that the advantage of the linear multigrid compared to the Jacobi preconditioner grows when reducing the number of fine level iterations. This suggests that the preconditioning effect of the coarse levels has a bigger effect relative to the required work. This is helped by the fact, that iterations on the coarse level are cheaper due to the reduced size of the linear system. In the late stages of convergence, when the density residual has reduced to below 10^0, the improved convergence properties of the linear multigrid preconditioner can be observed, apparent by the steeper slope of the curve. This can be on the one hand attributed to requiring fewer GMRES iterations to arrive at the requested residual reduction and on the other hand due to the fixed maximum number of Krylov iterations, which are not always able to reach the requested residual reduction.

4.3. Influence of Transfer Operator

In previous experiments, the restriction of the residual was done by a uniform summation of the residuals. This results in a restriction in which all sub-cells have equal influence on the residual of the coarse cell. Another approach to the residual restriction can be formulated by including mesh information into the restriction. This has been implemented in the form of a volume-weighted restriction, where each sub-cell residual contribution to the total residual of the coarse cell is weighted by its relative volume compared to the total volume of the coarse cell.

A comparison between these 2 transfer operators is performed for using the same smoother on the fine and on the coarse level. On each smoother step, 20 preconditioned Jacobi iterations are performed. The linear multigrid is compared to a classic Jacobi preconditioner with an equivalent number of Jacobi iterations. The results of the simulation can be found in figure 4.3



(a) (b)

Figure 4.3: (a) Number of GMRES iterations needed to arrive at the requested linear residual reduction for each non-linear iteration. (b) Convergence of the density residual plotted against wall-clock time.

In figure 4.3(a) the strength of the preconditioning is assessed, where a lower number of linear iterations indicates stronger preconditioning. From the figure, it can be seen that the linear multigrid methods need on average 4 times fewer iterations of the GMRES solver indicating a significant improvement in preconditioning strength. However, the volume-weighted transfer operator performs slightly worse than the uniform restriction in terms of preconditioning strength. In figure 4.3(b) the convergence history of the density residual is plotted. Similarly to the previous experiment, a performance benefit for the linear multigrid as a preconditioner can be observed. The uniform summation also performs slightly better than the volume-weighted restriction. This can not solely be attributed to the slightly better preconditioning, but also to the fact that the transfer operator for the volume-weighted residual restriction needs a proper Galerkin transfer for construction of the coarse level operator, which involves 2 matrix-matrix multiplications, which are not optimized in the code. A run time comparison of the Galerkin transfer was performed for the the construction of the coarse matrix using the uniform summation transfer, which showed an increase in runtime of approximately 6%.

Chapter 5

Conclusion

A non-linear multigrid method was implemented for two coarsening strategies, namely the coarsening using a mesh agglomeration and a coarsening which reduces the order of the discretization, by decreasing the maximum order of the basis functions used for the representation of the solution. These methods were also combined into a nested multigrid strategy, which combines the two coarsening strategies into a single multigrid method. Furthermore, a linear multigrid preconditioner was implemented for the use with second-order finite volume methods. This preconditioner is capable of working with general linear transfer operators and thus the effect of the transfer operator choice could be assessed. Both methods aim to reduce to computational time of complex CFD simulations, thus the results are interpreted mainly with regard to the decrease in computational time. This section will summarise the findings of the numerical experiments performed. First findings of the importance of the penalty parameter in the BR2 scheme are presented and their probable origin is discussed. In the following part, the results of the experiments performed for the non-linear multigrid methods for high-order discontinuous Galerkin (DG) discretizations are summarised. Lastly, the potential of a linear multigrid preconditioner in the context of Finite Volume (FV) methods is laid out.

5.1. The penalty parameter in BR2 on agglomerated meshes

Current literature suggests that the BR2 scheme is stable when choosing the penalty parameter η equal to the number of faces of the cell[5]. It suggests that on agglomerated meshes this is also applicable by choosing η as the number of facets of the cell. In practice however, it was seen that this choice did not lead to convergence for all test cases. It was observed that this classical choice works well for "viscous dominated" test cases such as the flat plate. However, when taking a look at flow situations that are dominated by convective flow features, this choice of penalty parameter leads to a stalling convergence. The employed linear solver is not able to solve the linearized system anymore. This however can be resolved by fixing η to a reasonably low value, e.g. 2. Previous investigations by Wallraff et al. [16] also noted that a rediscretization on coarse level lead to instability of the solution and also proposed that the treatment of viscous fluxes could be a problem. In their work, the problem was solved by using a Galerkin transfer for the coarse level operator.

This suggests an increase in stiffness of the Jacobian when using a facet based penalty parameter. However, due to time constraints, no eigenvalue analysis of the coarse Jacobian for a facet count penalty parameter in comparison to a fixed penalty parameter was performed so no conclusion could be reached about the origin of the observed problems. This suggests an increase in stiffness of the Jacobian when using a facet based penalty parameter.

5.2. Non-linear multigrid methods for high-order discontinuous Galerkin discretizations

From the performed experiments it could be seen that h and p non-linear multigrid methods can provide an improvement in all tested cases in terms of iteration numbers needed. For most cases, this also resulted in a reduction in computational time needed. This advantage was more pronounced for larger meshes. Furthermore, an increase in robustness to the initial CFL number can be noticed. When nesting different non-linear multigrid methods, these effects are conserved and in many cases stronger

than for simple multigrid methods. In practice, the use of a ph non-linear multigrid was found to perform at least equally well as all other methods and in most cases significantly better than those methods, while being able to converge with higher initial CFL numbers.

Due to the implementation, the startup of high-order computations could be simplified compared to previous methods. Furthermore, a nested startup procedure is possible for the computation of initial conditions for numerically very stiff problems. Such problems were not possible to be computed before in a reasonable amount of time.

5.3. Linear multigrid methods for a Finite Volume discretization

A linear multigrid was implemented into the existing infrastructure allowing its use as a preconditioner in the linear solver for Finite Volume discretizations. The implemented version can handle general transfer operators, which allowed the comparison of a uniform summation transfer operator to a volume-weighted transfer operator. It was found that for the considered test case, the linear multigrid preconditioner provided significantly stronger preconditioning compared to the current standard Jacobi preconditioner. This increase in preconditioning strength resulted in a reduction in computational time, for equivalent simulation parameters. It was furthermore shown, that there is a potential for performance improvement of the linear multigrid preconditioner, by shifting the relative work balance to the coarser levels.

The two aforementioned choices for the transfer operator were evaluated and it was found, that the volume-weighted transfer operator was not able to improve the behaviour of the linear multigrid preconditioner. Due to time constraints, no numerical simulations for the behaviour of this method in the case of high-order discontinuous Galerkin could be performed.

Chapter 6

Recommendations for future work

The developed linear and non-linear multigrid methods aim to reduce the computational time of current discontinuous Galerkin and Finite Volume methods. The improvement was shown for two-dimensional test cases. However it is unclear if these improvements transfer to more complex test cases. Future work should address these shortcomings. For non-linear multigrid methods, an assessment of the performance for three-dimensional test cases should be made, leveraging an improved version of the CoMMA agglomerator. Furthermore, a more detailed analysis of the problems encountered for the penalty parameter of BR2 on agglomerated meshes should be performed. As in the current work, the coarse level operator was constructed using a rediscretization for non-linear multigrid methods, the possibility for a Galerkin transfer should be explored.

The linear multigrid preconditioner should be tested on more industry-relevant test cases in three dimensions. Furthermore, an extension to high-order discontinuous Galerkin methods is possible.

Bibliography

[1] Steven R Allmaras, Forrester T Johnson, and Philippe R Spalart. "Modifcations and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model". en. In: *Computational Fluid Dynamics* (2012), p. 11.

[2] F. Bassi and S. Rebay. "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier–Stokes Equations". en. In: *Journal of Computational Physics* 131.2 (Mar. 1997), pp. 267–279. ISSN: 0021-9991. DOI: 10.1006/jcph.1996.5572. URL: https://www.sciencedirect.com/science/article/pii/S0021999196955722 (visited on 12/04/2021).

[3] F. Bassi and S. Rebay. "High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations". en. In: *Journal of Computational Physics* 138.2 (1997), pp. 251–285. ISSN: 00219991. DOI: 10.1006/jcph.1997.5454. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999197954541 (visited on 12/04/2021).

[4] F. Bassi et al. "Agglomeration based discontinuous Galerkin discretization of the Euler and Navier–Stokes equations". en. In: *Computers & Fluids*. "High Fidelity Flow Simulations" Onera Scientific Day 61 (2012), pp. 77–85. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2011.11.002. URL: https://www.sciencedirect.com/science/article/pii/S0045793011003367 (visited on 01/21/2022).

[5] F. Bassi et al. "On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations". en. In: *Journal of Computational Physics* 231.1 (Jan. 2012), pp. 45–65. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2011.08.018. URL: https://www.sciencedirect.com/science/article/pii/S0021999111005055 (visited on 01/21/2022).

[6] Francesco Bassi et al. "Discontinuous Galerkin solution of the Reynolds-averaged Navier–Stokes and k–w turbulence model equations". en. In: *Computers & Fluids* 34.4-5 (2005), p. 34. DOI: 10.1016/j.compfluid.2003.08.004.

[7] Ami Harten and James M Hyman. "Self adjusting grid methods for one-dimensional hyperbolic conservation laws". en. In: *Journal of Computational Physics* 50.2 (May 1983), pp. 235–269. ISSN: 00219991. DOI: 10.1016/0021-9991(83)90066-9. URL: https://linkinghub.elsevier.com/retrieve/pii/0021999183900669 (visited on 02/10/2022).

[8] Stefan Langer. "Agglomeration multigrid methods with implicit Runge–Kutta smoothers applied to aerodynamic simulations on unstructured grids". en. In: *Journal of Computational Physics* 277 (Nov. 2014), pp. 72–100. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2014.07.050. URL: https://www.sciencedirect.com/science/article/pii/S0021999114005439 (visited on 01/22/2022).

[9] T. Leicht et al. "DLR-Project Digital-X: Next generation CFD solver 'Flucs'". en. In: *undefined* (2016). URL: https://www.semanticscholar.org/paper/DLR-Project-Digital-X%3A-Next-generation-CFD-solver-Leicht-Vollmer/af310fd1a9c935d019638943a3c6db5543288919 (visited on 01/21/2022).

[10] Hong Luo, Joseph D. Baum, and Rainald Löhner. "A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids". en. In: *Journal of Computational Physics* 227.20 (Oct. 2008), pp. 8875–8893. ISSN: 00219991. DOI: 10.1016/j.jcp.2008.06.035. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999108003604 (visited on 01/23/2022).

[11] Wim A Mulder and Bram Van Leer. "Experiments with implicit upwind methods for the Euler equations". en. In: *Journal of Computational Physics* 59.2 (June 1985), pp. 232–246. ISSN: 00219991. DOI: 10.1016/0021-9991(85)90144-5. URL: https://linkinghub.elsevier.com/retrieve/pii/0021999185901445 (visited on 01/23/2022).

[12] W. H. Reed and T. R. Hill. "Triangular mesh methods for the neutron transport equation". English. In: Los Alamos Scientific Lab., N.Mex. (USA), Oct. 1973. URL: https://www.osti.gov/biblio/4491151-triangular-mesh-methods-neutron-transport-equation (visited on 02/06/2022).

[13] P. Spalart and S. Allmaras. "A one-equation turbulence model for aerodynamic flows". en. In: *30th Aerospace Sciences Meeting and Exhibit*. Reno,NV,U.S.A.: American Institute of Aeronautics and Astronautics, Jan. 1992. DOI: 10.2514/6.1992-439. URL: https://arc.aiaa.org/doi/10.2514/6.1992-439 (visited on 01/23/2022).

[14] U. Trottenberg, C. W. Oosterlee, and Anton Schüller. *Multigrid*. en. San Diego: Academic Press, 2001. ISBN: 978-0-12-701070-0.

[15] Michael Wagner et al. "Spliss: Transparent Integration of Heterogenous HPC Architectures into CFD Solvers and Applications". en. In: *Methods, Tools and Technologies for Design in Aviation*. Barcelona, Spanien, Nov. 2021. URL: https://elib.dlr.de/147266/ (visited on 01/21/2022).

[16] Marcel Wallraff and Tobias Leicht. "Higher Order Multigrid Algorithms for a Discontinuous Galerkin RANS Solver". en. In: *52nd Aerospace Sciences Meeting*. National Harbor, Maryland: American Institute of Aeronautics and Astronautics, Jan. 2014. ISBN: 978-1-62410-256-1. DOI: 10.2514/6.2014-0936. URL: https://arc.aiaa.org/doi/10.2514/6.2014-0936 (visited on 01/21/2022).

DLR-IB-AS-BS-2021-227

H- and P-Multigrid Methods for Higher-Order

Discontinuous Galerkin Discretizations

Malte Wegener

Verteiler:

Institutsbibliothek	1 Exemplar
Verfasser	3 Exemplare
Institutsleitung	1 Exemplar
Abteilungsleiter	1 Exemplar
Deutsche Bibliothek in Frankfurt/Main	2 Exemplare
Niedersächsische Landesbibliothek Hannover	1 Exemplar
Techn. Informationsbibliothek Hannover	1 Exemplar
Zentralbibliothek BS	2 Exemplare
Zentralarchiv GÖ	1 Exemplar
Reserve	2 Exemplare